An Evaluation Framework for Stereo-Based Driver Assistance

Nicolai Schneider[1], Stefan Gehrig[2], David Pfeiffer[2], and Konstantinos Banitsas[3]

[1] IT-Designers GmbH, Esslingen, Germany
[2] Daimler AG, Team Image Understanding, Sindelfingen, Germany
[3] Brunel University, London, UK
nicolai.schneider@it-designers.de
{stefan.gehrig,david.pfeiffer}@daimler.com
konstantinos.banitsas@brunel.ac.uk

Abstract. The accuracy of stereo algorithms or optical flow methods is commonly assessed by comparing the results against the Middlebury database. However, equivalent data for automotive or robotics applications rarely exist as they are difficult to obtain. As our main contribution, we introduce an evaluation framework tailored for stereo-based driver assistance able to deliver excellent performance measures while circumventing manual label effort. Within this framework one can combine several ways of ground-truthing, different comparison metrics, and use large image databases.

Using our framework we show examples on several types of ground-truthing techniques: implicit ground truthing (e.g. sequence recorded without a crash occurred), robotic vehicles with high precision sensors, and to a small extent, manual labeling. To show the effectiveness of our evaluation framework we compare three different stereo algorithms on pixel and object level. In more detail we evaluate an intermediate representation called the *Stixel World*. Besides evaluating the accuracy of the Stixels, we investigate the completeness (equivalent to the detection rate) of the Stixel World vs. the number of phantom Stixels. Among many findings, using this framework enables us to reduce the number of phantom Stixels by a factor of three compared to the base parametrization. This base parametrization has already been optimized by test driving vehicles for distances exceeding 10000 km.

1 Introduction

Today's stereo and flow algorithms have reached a maturity level that allows their use in real-world systems. The development of efficient stereo algorithms is the first step in making vehicles able to recognize their surroundings and eventually drive themselves in the future. Unfortunately, the performance evaluation for such algorithms is still mostly limited to comparisons on the Middlebury database[1] [32]. There, stereo and flow algorithms are benched against a few indoor images under controlled conditions. Most applications have to deal with a

[1] e.g. http://vision.middlebury.edu/stereo/

lot of different conditions which are not covered by such controlled data sets. Especially in the automotive field a limited sensitivity to adverse weather conditions is crucial. This requires a certain robustness of the applied algorithms. For such an outdoor imagery evaluation we need metrics to evaluate different algorithms or parameters and to compare their performance. The goal is to create a system that will automatically evaluate the computed 3D scene description of the environment. For this purpose, we introduce a performance evaluation framework considering the following three levels:

1. low-level: *pixel-level*, e.g. false stereo correspondences - based on stereo data where we use knowledge about object-free volumes to detect violations.
2. mid-level: *freespace/Stixel* [2], the object-free space in front of the car — the inverse is also called evidence grid/occupancy grid. This is computed directly from the stereo correspondences. The freespace forms a basis for many other object detection algorithms and thus is suitable for a mid-level evaluation. Similar, the Stixel World describes the objects limiting the freespace and is evaluated in detail here.
3. high-level: *leader vehicle measurement.* We pick one particular application where the leading vehicle is measured in front of the ego-vehicle. This data is needed for all adaptive cruise-control (ACC) variants. Depending on the implemented driver assistance function, different accuracy demands are needed for the distance, relative velocity, lateral position and width of the leading vehicle. We focus on the lateral position and width of the leader vehicle since we have a RADAR system that determines the distance and relative velocity very accurately and serves as ground truth for that part. The challenge for such applications is to create a correct object segmentation, and it is here that the choice of stereo algorithm becomes apparent.

Our evaluation framework working on these three levels covers the range of applications in which stereo is used in today's automotive industry (e.g. [37]).

The structure of this paper is as follows: The related work on our system is detailed in Section 2. The basic framework for this analysis is described in Section 3. The ground truth needed to evaluate the tasks is introduced in the same section. To show the power of the evaluation framework we select several algorithms for evaluation that are described briefly in Section 4. In Section 5 more details on the used metrics to measure the performance are given. We have tested three different stereo algorithms on all evaluation levels of detail and show the results in Section 6.1. Evaluation results focusing on several aspects of the Stixel World are presented in Section 6.2.

2 Related Work

2.1 Evaluation of Computer Vision Algorithms

In the field of automotive, computer vision systems become increasingly powerful. Consequently, many driver assistance systems make use of them for. However, under adverse weather conditions these systems do not posses the reliability required. Using image based sensor information for active braking or autonomous steering requires high safety levels, robustness, and accuracy with respect to the used algorithms. The more safety critical a system is the more effort has to be spent in the evaluation process of such vision algorithms. The correctness and the required integrity of these systems gain special importance when upcoming norms like ASIL (Automotive Safety Integrity Levels or ISO 26262) come into effect.

In [27], a general framework for performance evaluation of Computer Vision algorithms is presented, with a focus on object detection algorithms. However, all introduced metrics are limited to monocular sequences and to metrics within the image plane. Both methods are less relevant to robotics and driver assistance scenarios.

One of the major problems in evaluating computer vision algorithms is the generation of ground truth data against which results can be tested. Traditionally, most of the algorithms in literature are evaluated by measuring differences between the computed result and the Middlebury database [44]. However, for automotive applications this is not sufficient, because the automotive field is faced with a couple of challenges: Firstly, it has to deal with adverse weather and lighting conditions which are not covered by such controlled data sets. Secondly, the tremendously rising complexity of modern vision systems demand for new evaluation methods which cannot be performed on single images. In addition, a pixel-by-pixel comparison (as on Middlebury) is not applicable to sparse stereo or flow algorithms - an algorithm class that might serve driver assistance tasks very well.

In general, algorithms need to be tested on much larger datasets for obtaining statistically meaningful performance measure [9]. A step towards creating large ground truth datasets was made in [18]. The authors presented a reliable methodology for establishing a large database of ground truth data for a variety of sensors on mobile platforms. The goal was to publish large datasets to support other researchers to verify and evaluate their algorithms. An evaluation strategy for stereo algorithms on large amounts of images was also proposed in [36]. In that publication a performance evaluation scheme and corresponding metrics were suggested. The authors describe a method for producing low effort evaluation results without having real ground truth data. Some of the obtained results are reiterated in this research.

2.2 Ground-Truthing

In recognition tasks (e.g. [10,12]) manually annotated ground truth is widely used where Receiver-Operator-Curves (ROC), Precision-Recall-Curves, or classification rates are compared. There, ground-truthing is already necessary to provide the recognition algorithms with training data.

An example used to easily obtain some ground truth data is shown in [24], where an orthogonal method to determine the street plane is used to evaluate stereo algorithms. However, the street plane investigation only verifies small parts of the image whereas for real automotive applications there are many other parts of the 3D scene which are of high importance.

In the current literature, several concepts have been used for generating ground truth data. Each of them have their corresponding advantages and drawbacks. The following sections will give a short overview of those concepts.

Multi Sensor Technology Modern test vehicles are usually equipped with multiple sensors. LIDAR (Light Detection And Ranging), RADAR (Radio Detection and Ranging) and optical cameras are examples for those sensors. Using a multi sensor system has the advantage of detecting (or even compensating) for the various errors produced by each method yielding more reliable and accurate data. Different approaches have been found which propose an efficient fusion strategy as well as solutions in handling divergent data [11,17,42].

In [31] an evaluation strategy for the Stixel World was published using a high precision LIDAR as a reference sensor. The Stixel's distance information was compared against the LIDAR measurements. Different scenarios were recorded and the errors in various distances were analyzed. In order to realize the proposed concept with low effort, the technical challenges in synchronizing the different sensors were circumvented using the stop motion principle. Leaving, only simple scenarios (without any dynamic driving maneuvering) can be analyzed.

According to [31], Semi-Global Matching (SGM) and LIDAR behave differently to reflective vehicle objects like windows, mirrors or puddles. While the SGM stereo estimation smooths over these areas, the LIDAR looks right through those or even follows the reflected rays of light: an undesirable property of such a system. Consequently, using LIDAR as ground truth sensor makes an evaluation in these areas impossible.

Another evaluation example using several sensors of the same type was published in [28]. In this approach various common stereo matching algorithms were evaluated using three cameras. Two of them were used for the stereo matching and the third was used for reference in order to estimate the prediction error. By using metrics assessing the intensity differences of the first two cameras and comparing those with the output of the virtually computed third camera, it was possible to rate different stereo algorithms on real-world scenes.

Manual Labeling Methods One of the most commonly used methods in generating ground truth data is the involvement of human expert interactions called *labeling*. As every application or algorithm has different requirements numerous approaches exist in designing ground-truthing tools. In general these can be categorized as automatic or semi-automatic ones [19]. The majority of the tools are semi-automatic as in most cases some additional information is needed for starting the ground truth extraction.

Tools supporting manual input often have the advantage that errors raised from model approximations or noisy data can be minimized through human verification and correction. These semi-automatic tools are not very efficient in generating large ground truth datasets as they involve human effort during the process.

Driver assistance imagery exhibit highly dynamic driving scenarios and often at least 50 frames are necessary in order to make a reliable statement on the performance of the applied algorithm. Consequently, labeling large amounts of sequences is time consuming. To overcome this problem some approaches were published incorporating available tracking mechanisms in ground-truthing tools [18]. Instead of labeling each frame from the beginning, trackers can be used to follow objects from one frame to the next so human inputs or corrections are only required if deviations occur [26].

Synthetic Data Today a lot of effort is put in generating realistic synthetic scenes. Based on a physical model, static and moving objects are rendered and placed into a defined scenario.

Using synthetic sequences has the advantage that all parameters for every object are previously known. This accounts especially for the trajectories of the moving objects. Hence, an evaluation becomes simple because ground truth data can be calculated from ray-tracing principles and thus is available for every single image of the sequence. Moving the viewpoint of a virtual image makes it possible to generate image pairs for simulating stereo-vision and computing the ground truth disparity image.

The drawback of using synthetic scenes is the increased entropy of real life: it is next to impossible to create models for all the real-world situations. Adverse weather conditions such as rain, sun glare or snow are examples of that as their physical background is too complex for mapping it to a computer model.

Study [40] shows how synthetic scenes can have an negative impact on the performance of stereo and motion estimation. Their results show that optimizing algorithms for synthetic data can even make the results on real-world scenes worse. For example motion blur, weather, and exposure differences between the left and right image can highly influence the performance of the algorithms.

3 Evaluation Framework

The main aim of our evaluation framework is to provide an automatic method for evaluating and optimizing different stereo and flow algorithms over a large dataset [36]. By now, it has proved its strength to be well suited for all kind of image processing tasks. A large sequence database with more than 1500 sequences (200-400 frames per sequence) serves as input for the evaluation task. Since most of the vision algorithms consume a lot of computing power the idea is to write the raw data measurements into an Evaluation Database (EDB) and calculate the metrics afterwards. This has the advantage that a recalculation of our metrics can be done within seconds. Figure 1 shows an overview of the framework.

All algorithms that are tested perform their image processing tasks with a predefined parameter set on the stored test sequences. For a meaningful eval-

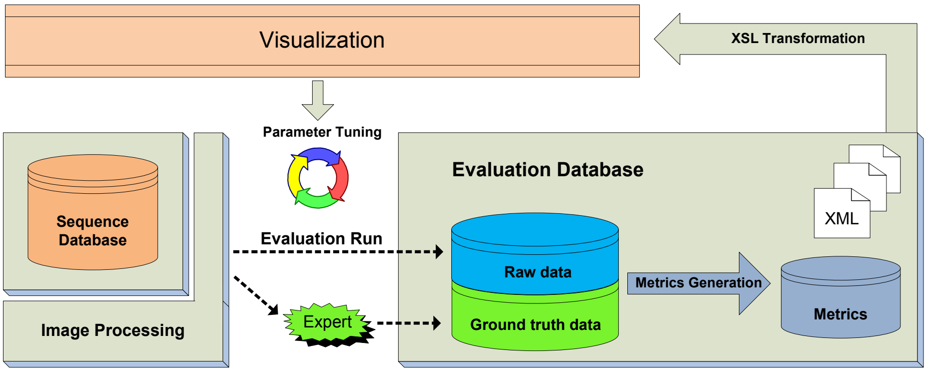


Fig. 1: Overview of the evaluation framework

uation, the content of the database has to differ with respect to daytime (day, night), weather (rain, snow, fog), location and environment (city, rural roads, highway). In an Evaluation Run (ER) for each frame of the sequence the measured raw data is written into the EDB. The ground truth data against which will be tested is either collected during a specific test run or defined manually by experts (manual ground-truthing). In case of manual ground-truthing, an appropriate software module is used providing manual interactions with the image. As a result of the image processing task a dataset with *ground truth* data and *measured* raw data is available in the EDB. A Metrics-Generator C++ module uses the generated datasets, computes the user defined metrics from it and saves it as an XML file back into the database. The processed data is visualized in a browser front-end by transforming the XML files with a predefined XSLT (http://www.w3.org/TR/xslt) style-sheet to SVG images. The transformation language (XSLT) provides an efficient strategy to transform a huge number of measurements into a few compact and easily explorable representations.

In addition, for each sequence a score is extracted by integrating the metrics frame-wise. By means of color encoded rankings one can easily determine those sequences which are relevant for further algorithm improvements. The user employs the sequence-wise accumulated metrics to choose candidates which could outperform the *current* ground truth. It takes only seconds in order to find and inspect relevant frames and to decide if the current candidate is a better ground truth or not. In order to verify the automatic testing process we use a subset of about 20 manually labeled ground truth datasets. A 3D editor and a tracking mechanism [4] allows effortless labeling of the scene infrastructure for this subset of sequences (see Figure 2). The accuracy of the manual ground truth is about 0.05 m error on average in the considered range ($0\,m$ - 40 m).

The 3D editor displayed in Figure 2 is used to create artificial ground truth data. For this purpose, static scene content from recorded sequences is projected into a virtual 3D view. Within that view, scene geometry is defined using basic

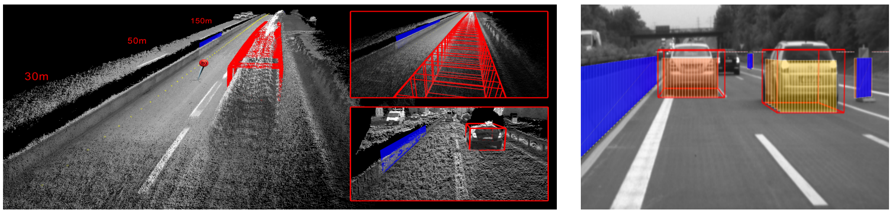


Fig. 2: A 3D editor is used to manually create ground truth scene data. The right image shows the corresponding 2D output. The blue walls describe static scene infrastructure and the red boxes result from an object tracking algorithm to effectively evaluate moving objects.

geometrical shapes. During this step, dynamic scene content is taken into account by using the boxed-based tracking scheme proposed by Barth et al. [5,6].

An additional source of ground truth are robotic vehicles operated on a proving ground. For accuracy evaluation these robotic vehicles (having high-precision IMU) are used to perform predefined maneuvers repeatedly with high accuracy (errors $< 0.02\,\mathrm{m}$). This results in accurately known position and motion states of the observed vehicles.

4 Algorithms Used

4.1 Stereo Algorithms

The initial motivation to build the evaluation system was in order to compare the following three stereo algorithms. All of these algorithms have real-time processing capability.

- *Signature-Based Stereo*: A signature based algorithm that searches for unique (corresponding) features of pixels [35].
- *Correlation Stereo*: A patch based correlation stereo algorithm using ZSSD (zero-mean sum of squared differences) [13].
- *Semi-Global Matching (SGM)*: Computes an approximated global optimum via multiple 1D paths in real-time [16].

4.2 Stixel World

The Stixel World [3,30] is a compact medium-level representation that describes the local three-dimensional environment. Stixels are defined as earthbound and vertically oriented rectangles with a fixed width (e.g. 5 px) and a variable height. Under these restrictions, Stixels are a 2.5D representation similar to Digital Elevation Maps [8]. From left to right, every obstacle within the image is approximated by a set of adjacent Stixels. This way, Stixels allow for an enormous reduction of the raw input data, e.g. 400.000 disparity measurements ($1024 \times 440\,\mathrm{px}$ stereo image pair) can be reduced down to only 200 Stixels.

Stixels simply give access to the most task-relevant information such as freespace and obstacles. For providing multiple independent vision-tasks with stereo-based measurement data, the Stixel World is neither too object-type specific nor too general and thus efficiently bridges the gap between low-level (pixel-based) and high-level (object-based) vision.

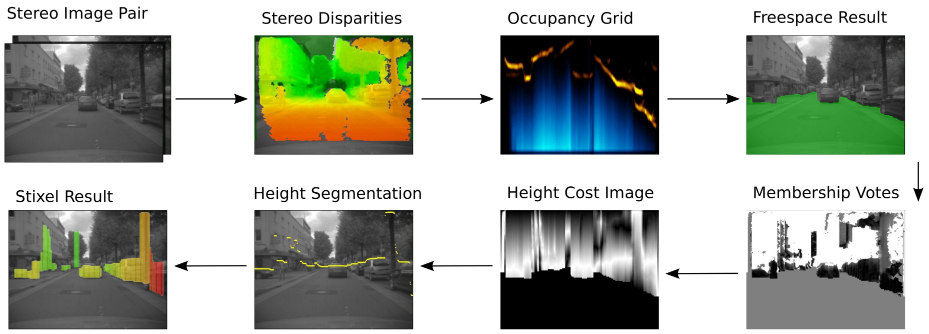


Fig. 3: The Stixel World is extracted from stereo data in a cascade of multiple processing steps. This includes stereo matching, mapping stereo data to occupancy grids, freespace computation, a height segmentation and the final Stixel extraction step.

According to [3], Stixels are computed in a cascade of multiple processing steps: Mapping disparities to occupancy grids, a freespace computation, a height segmentation, and the final Stixel extraction step. For clarity, that process is visualized in Figure 3. Besides using Stixels to represent static environments, relying on the 6D-Vision [14] based Kalman filtering techniques allows for robustly tracking Stixels over time. Since the tracked objects are expected to move earthbound, the estimated state $\underline{X}$ is four-dimensional and consists of the lateral (X) and longitudinal (Z) position as well as the corresponding velocity components, such that $\underline{X} = (X, Z, \dot{X}, \dot{Z})^T$. As a result, motion information about the obstacles in the scene is available for every Stixel independently [30].

Making this work demands to have knowledge about the own motion state and requires to measure displacements of Stixels between two consecutive images. Ego-motion is provided by either visual odometry [1,15], SLAM [21,22,23] (self localization and mapping), or the inertial motion sensors of the vehicle. Stixel motion is obtained by computing optical flow correspondences. To achieve this a number of different methods are described in current literature. A short selection of those methods is listed below.

4.3 Optical Flow Schemes

Tracking Stixels over time in order to estimate the velocity of other moving obstacles requires the measuring of the two-dimensional displacement of these objects within the images of two consecutive time steps. This is achieved by computing the optical flow correspondences for exactly those areas.

Within the scope of this evaluation, four different flow methods have been chosen for testing. In the following, their particular differences, assets and drawbacks are highlighted briefly.

Sparse KLT In [25], Lucas et al. suggest an optical flow scheme for feature tracking that relies on the gradient-based Lucas-Kanade method. The actual displacement is computed by solving the optical flow equations resulting from the constant brightness assumption for all the pixels in the neighborhood of a center point. This is achieved by means of a least squares error minimization.

Aiming at gaining robustness to global illumination changes, the matching criteria of this scheme is adapted to support a more robust measure, the zero-mean sum of squared differences (ZSSD).

A benefit of this method is the possibility to use the Kalman filter prediction of the tracked objects for initialization. This noticeably supports the estimation of large optical flow vectors and reduces effects resulting from texture ambiguities (e.g. repetitive patterns such as guard rails).

Patch KLT The Patch KLT method is an extension of the KLT feature tracker to larger $m \times n$ sized feature patches. In order to take perspective effects into account the change of scale is part of the estimation process. Additionally, an individual weight is considered for each pixel that is computed from the corresponding disparity measurement and the disparity of the tracked Stixel. This way, the influence of (background) pixels that lie within the patch (but do not belong to the actual tracked object) is minimized.

The Patch KLT benefits from leveraging texture information much better than competitive methods. Just like the sparse KLT method, the Patch KLT allows to be initialized with the prediction of the Kalman filter state.

Census Optical Flow Stein [35] presents a method that allows to compute optical flow using the Census transform [41] as matching criteria. The census signatures are mapped to a hash table which is then used to determine optical flow correspondences between two images.

The benefit of this method is the constant run time independent of the maximum optical flow vector length. On the downside, this scheme does not allow to incorporate the motion state of the tracked object during initialization.

Dense TV-L1 Optical Flow Müller et al. [29] have proposed a dense TV-L1-based method that puts dedicated focus on the application in open road scenarios. It incorporates additional stereo and odometry knowledge about the three-dimensional scenario. Their scheme is a variant of the work proposed by Zach et al. [43]. The implementation used does not consider information about the objects motion state for initialization.

5 Used Evaluation Metrics

Evaluating over large datasets demands effortless execution strategies and simple metrics which yield valuable information on the robustness and accuracy of an algorithm. Low-level metrics reflect the performance of a pixel-wise algorithm (e.g. the stereo matching scheme), mid-level metrics rate the quality of a possible intermediate representation at a later stage (e.g. the Stixel World) in the data processing chain, and high-level metrics consider the object level. The used metrics are described in more detail in [33].

Typically, errors occur on sensor failures, atypical events (e.g. wipers crossing the windshield), or adverse weather and poor lighting conditions. Thus, for the purpose of our evaluation the following two aspects are examined:

- *Robustness:* represents the algorithm's ability to deal with challenging situations like adverse weather and lighting conditions.
- *Accuracy:* describes the precision with which a Stixel represents the object in the real world.

5.1 Robustness

In the context of safety-critical vision-based driver assistance, the robustness of the used algorithms is of uttermost importance. With respect to robustness, it is reasonable to distinguish between algorithms operating on the pixel layer and those that use the object layer. For instance, a single error during stereo matching is rather unlikely to lead to a drastic automatic intervention of the driving car. However, the situation might be different for several false alarms in the medium-level representation.

Naturally, object occurrences in the driving corridor have a high priority, because those objects might lead to a critical change in driving. Hence, the evaluation primarily focuses on errors occurring within the driving corridor.

False Stereo Correspondences (low-level) When dealing with a large sequence database it is neither practical nor expedient to create ground truth data manually. This is especially true for disparity depth maps, as this method turns out to be a very time-consuming and hardly a feasible undertaking. In our research, a different strategy was chosen:

The vehicle's driven path through the three-dimensional scene is reconstructed before the evaluation. This is achieved by looking ahead the vehicle's odometry information (velocity and yaw rate) from the recorded sequence meta-data. It enables us to evaluate the false positive rates up to distances of 40 m. In case of having other moving vehicles in the scene, the actual freespace is additionally restricted by using an independent RADAR sensor (Continental ARS300 long range RADAR [34]). During this process, the RADAR is considered as ground truth and the RADAR results were checked visually by backprojecting the RADAR results into the image. For clarity, the described strategy is illustrated in Figure 4.

Given good visual conditions, no stereo measurements should fall into that volume. Hence, all stereo correspondences that do so are registered as potential errors of the stereo matching scheme. Following that strategy allows us to process many sequences without the need for human inspection or interaction. In return, that gives us the opportunity to evaluate very large sequence data bases with minimal effort.

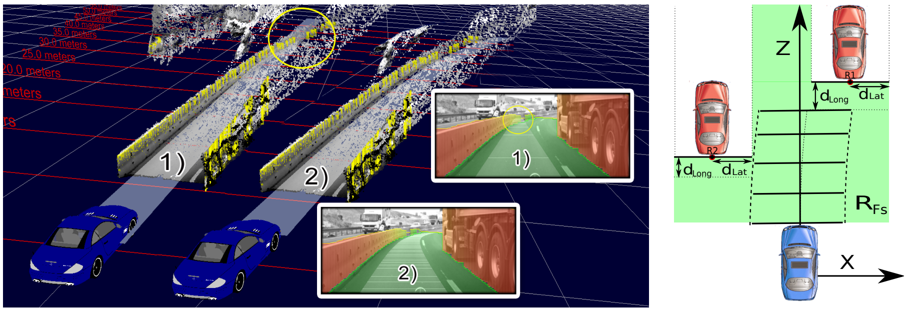


Fig. 4: With a prediction of the ego vehicle's current state it is not possible to detect the oncoming sharp right-hand bend early enough (marked with the yellow ellipse). That means that only a prediction around 25 m would be possible. Instead, by looking ahead the odometry information and RADAR information allows us to define the drivable freespace up to distances of 40 m. The diagram on the right side depicts such a freespace.

False Positives (mid-level) In order to test for false positive detections on the medium level, the same strategy (as above) is followed.

In terms of the Stixel representation, a false positive is defined as a Stixel detection that cannot be associated to an actual object in the real world. Thus, similar to detecting false stereo correspondences, all Stixel observations that lie within the driving path are considered as false detections.

Detection Range Another important characteristic for a vision system is the achieved detection range. Thus, for judging this property adequately, a so called completeness measure is defined. It reflects the detection rate of objects in the scene using ground truth object data.

For this evaluation task we use two different types of input data: Manually labeled sequences with a known 3D world geometry as well as robotic sequences. Robotic sequences correspond to driving scenarios with automated vehicles of which one carries the stereo system. Their motion path is known precisely by using iMAR iTrace-RT-F200 [20] IMUs as well as differential GPS.

For each time step in this database a corresponding ground truth Stixel representation is computed. A particular ground truth Stixel is considered as detected if a corresponding Stixel is computed from the input images (true positive). Consequently, both Stixels have to be within a depth-range of ±1 m or ±3 px disparities. Otherwise, the object is considered as missed (true negative). The corresponding completeness measure is defined as the ratio of the number of detected Stixels over the expected total amount of Stixels.

Colliding Stixels In order to determine the robustness of the Stixel velocity estimate, it is preferable to have real-world ground truth data for all moving objects in a scene. Again, this is hard to achieve for a large dataset so instead of performing a direct comparison, we use the Time To Collision (TTC) of a Stixel as an indicator for tracking errors. Since all of the scenarios in the database are recorded without having a collision, it can again be assumed that the TTC to other objects (static and moving) is greater than 1 s. Hence, if the predicted position of a Stixel intersects with the predicted vehicle position, a tracking error is registered. Figure 5 shows an exemplary inner city scenario. The red area visualizes the ego vehicle's position within the next second. The arrows on the ground plane denote where the Stixels will move in that same period of time.

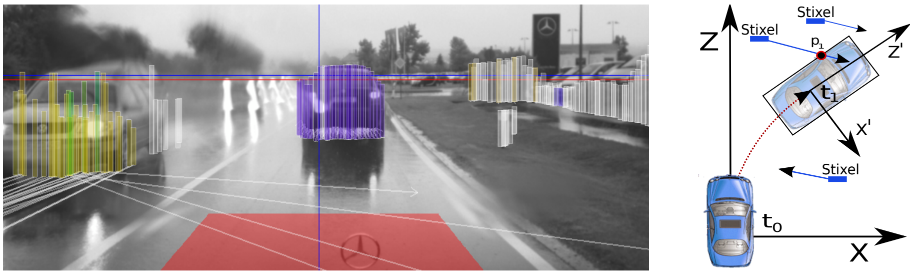


Fig. 5: Exemplary inner city scenarios with colliding Stixels. The red carpet indicates the ego vehicle's position in the next second. The white arrows denote the Stixel positions in the same period of time. On the right the intersection check is visualized.

5.2 Evaluation of Accuracy

For accuracy evaluation the robotic vehicles are used. The vehicles perform predefined maneuvers. The IMU units record the exact paths of both platforms. The data is used for testing the accuracy of the distance measurement as well as the precision of the estimated velocity.

Distance Error Both IMU units provide an accurate motion state for every frame. Using this data allows to transform all robotic motion states into the ego-system of the stereo camera rig used for testing. From that point onwards, it is straightforward to extract all Stixel measurements that are located on the other vehicle's front and determine their mean distance so. This value is compared to the ground truth data of the IMU units.

Velocity Error The evaluation of the velocity tracking error is split in two parts. Firstly, under the assumption that the current sequence is recorded in a static environment (i.e. without any moving objects), the mean absolute velocity error over all Stixel velocity estimates should equal to zero. Secondly, to evaluate while dealing with moving objects the robotic sequences are used. This way, the IMU velocity data is compared to the mean velocity estimate of those Stixels that represent the vehicles front in the image.

6 Evaluation Results

6.1 Stereo Performance

The sequences we used for stereo evaluation are divided into 50 % bad weather conditions (rain, snow, night) and 50 % normal conditions and contain a total of 22.100 frames recorded at 25 fps. The mixture is chosen to find failure modes of the algorithms as quickly as possible so less data will be needed.

The results in Table 1 show that the Signature-Based Stereo exhibits some shortcomings. Correlation Stereo is far better than Signature-Based Stereo, but the best method at all levels and metrics is SGM. The results from the bad weather part of the database are shown separately using parentheses.

Furthermore, the freespace computation and the leader vehicle measurement parameters were tuned for the Correlation Stereo method. For this reason, the obtained results underline the overall good and stable performance of SGM. If the applications were tuned with respect to SGM, the results of SGM would be even better. Especially the *availability* of the leader vehicle measurements outperforms the correlation approach by far. With the used freespace algorithm and metric at hand, we obtain similar results and the same ranking of stereo algorithms using the Stixels as intermediate representation.

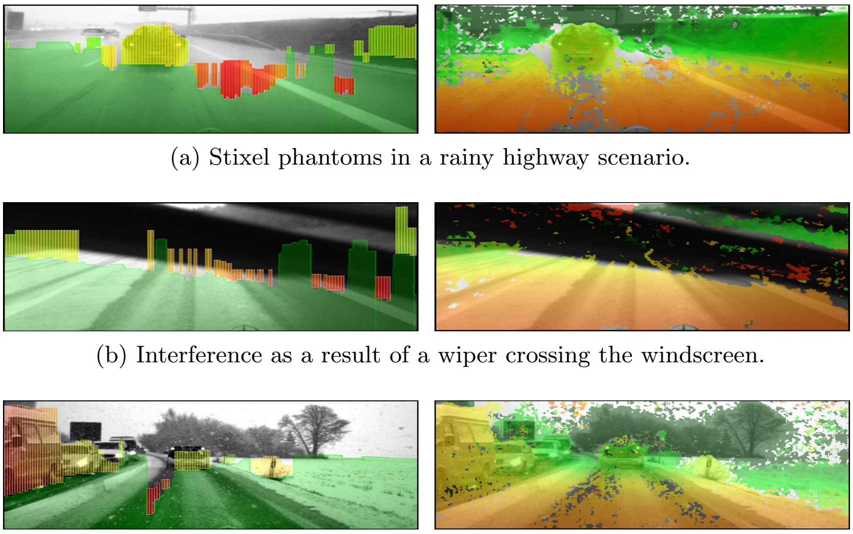
Metric	low-level	mid-level	high-level (*LVM*)		
	False Corr.	*FS Diff.*	*ΔLat.Pos.*	*ΔWidth*	*Availability*
Algorithm / Unit	m_{fc} [%]	m_{fs} [px]	m_{lp} [cm]	m_w [cm]	[%]
Signature-Based Stereo	7.45 (10.35)	3.04 (3.06)	19 (22)	26 (35)	80 (88)
Correlation Stereo	1.02 (1.47)	1.26 (1.72)	13 (15)	19 (37)	95 (99)
Semi-Global Matching	0.98 (0.94)	0.68 (0.79)	11 (12)	14 (32)	99 (99)

Table 1: Evaluation result comparing census-stereo, correlation stereo, and SGM. SGM outperforms the other algorithms on all levels of detail.

6.2 Stixel Robustness

For the robustness evaluation of the Stixels, the complete database with more than 500 recorded sequences was used. It includes typical urban environments, rural roads and highway scenarios at different day times and weather conditions.

Phantom Rate The Stixel phantom rate was determined in the categories *Sunshine*, *Night*, *Rain*, *Heavy rain* and *Snow* and is measured in phantoms per frame. Examples of challenging scenes with occurring phantoms are shown in Figure 6.



(a) Stixel phantoms in a rainy highway scenario.

(b) Interference as a result of a wiper crossing the windscreen.

(c) Snow scenario with phantom Stixels.

Fig. 6: The depicted Figures show different challenging scenarios of failure cases for the stereo computation and thus for the Stixel extraction. The visualization shows both the freespace/Stixel result as well as the disparity image.

The results in Figure 7 primarily show that under optimal environmental conditions an excellent low error rate is achieved. However, this result change for adverse weather conditions such as *Rain* or even *Heavy Rain*, where the phantom rates are considerably higher. *Snow* on the other hand turns out to be less of a problem than anticipated. This effect is mainly explained by the fact that, in contrast to rain, snow does not necessarily lead to a wet windshield and therefore does not cause a blurred sight.

In order to optimize for a low phantom rate, different parameters of the Stixel extraction schemes have been fine-tuned. At the same time it was important to consider the completeness metric. Otherwise, minimizing the phantom rate

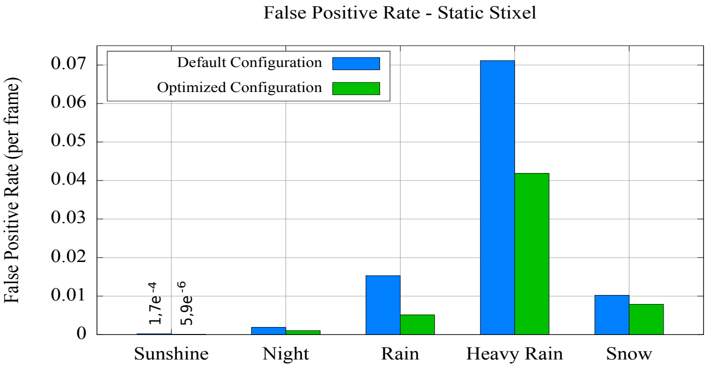


Fig. 7: The diagram depicts the Stixel phantom rates in the categories Sunshine, Rain, Heavy rain and Snow evaluated on a dataset exceeding 500 sequences.

would inevitably lead to an arbitrary and possibly undesirable reduction of the object detection rate. In the sense of an ROC-curve, this dependency is visualized in Figure 8. The optimization was performed using manually labeled ground truth sequences with available 3D world geometry. This database consists of 20 manually labeled sequences with a total sum of approximately 1000 objects. The images in Figure 8 illustrate an extract of labeled database objects (red is moving, blue is static). In addition, the diagram depicts the limit up to which an optimization allows a robust object detection. A 100 % detection rate can not be reached due to violations of the assumed vertical pose constraint.

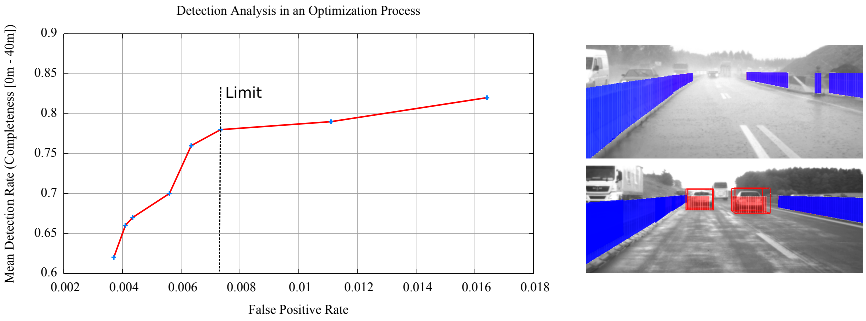


Fig. 8: The figure visualizes the completeness measure averaged over all labeled objects in the database. With an increasing optimization level, the number of phantoms decreases. However, a small phantom rate results in a low completeness. The diagram shows the limit after which further optimizations would downgrade the detection rate too much.

Detection Range The detection range was evaluated on robotic scenarios. The priority was on that distance where the object detection exceeds 90 % completeness on the robotic vehicle. Consequently, for this purpose, only scenarios with an oncoming vehicle covering a range of 0 m − 80 m were of interest. Figure 9 shows the completeness over the distance. In order to have a meaningful result, the completeness is averaged over several sequences of the same type.

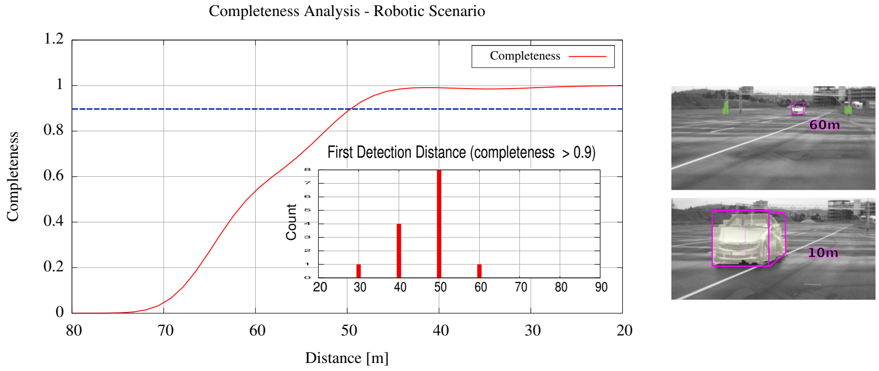


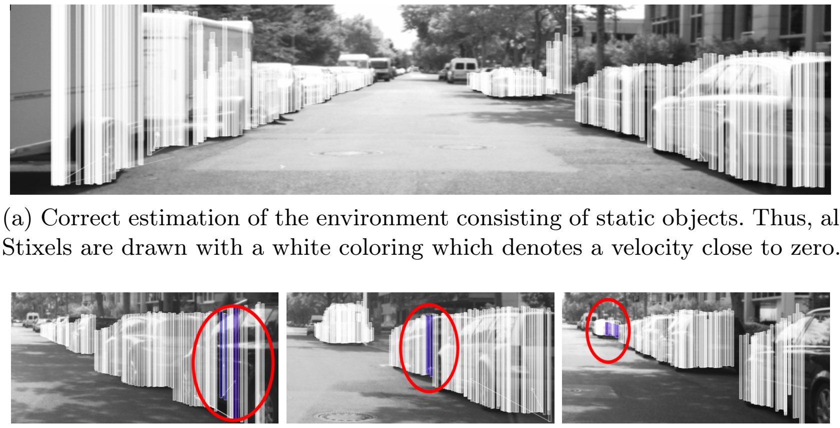
Fig. 9: The diagram shows the completeness in relation to the distance. The embedded histogram show the distribution of the distances at which the robotic vehicle reaches the 90 % completeness level for the first time.

6.3 Tracking Performance - Testing Different Optical Flow Methods

Within Section 4.3 different Stixel tracking strategies have been discussed. This section aims to evaluate their performance and their quality with respect to the estimated motion states.

In terms of the object tracking, a core aspect is the computation of the Stixel displacement between the two consecutive images. For that task, we discussed multiple approaches that differ with respect to their technical prerequisites, their scope of action to combine the optical flow computation directly with the actual tracking process, and their computational effort. The Stixel tracking was tested in a stationary environment that contained no moving objects. Even though our own car was moving, the goal was to detect that the environment around us remains static.

The test using a static environment took place in a narrow urban environment with cars parked on both sides of the road. Naturally, the expectation for the motion state of all tracked objects is to have zero velocity. To stress the optical flow methods, the scenario was recorded several times while driving at different speeds, which includes 4, 8, 14 and 20 $^{\mathrm{m}}/_{\mathrm{s}}$ ego-velocity. A snapshot of that sequence is depicted in Figure 10. The given figure also discusses different challenging aspects for the optical flow computation.



(a) Correct estimation of the environment consisting of static objects. Thus, all Stixels are drawn with a white coloring which denotes a velocity close to zero.

(b) Three typical sources for velocity errors during tracking within static environments are illustrated. The first figure shows a reflecting surface, the middle figure shows a jump in depth, and the third figure shows difficulties with motion estimation at large distances.

Fig. 10: Color visualizes motion. Ideally, all static objects should have a white coloring denoting zero velocity. This real-time color coding was used as quality indicator for the different tracking schemes. Figure (a) shows a good example, Figure (b) shows typical sources of error.

For estimating the optical flow between consecutive time steps the *Census-based* feature flow proposed by Stein [35], the dense *TV-L1* based optical flow scheme proposed by Müller [29], the *KLT-based* feature tracker proposed by Tomasi [39] and our own *Patch KLT* method were used. For the latter, a patch size of 40×16 px (width×height) has proved a good working choice.

The results for the static environment are depicted in Figure 11. On the left side, for rating the tracking performance of the individual tracking schemes, the mean absolute velocity of all tracked Stixels is computed. Depending on the ego-velocity of the test vehicle, each sequence contributes about 300 to $1,000$ frames. The evaluation is limited to a distance of 40 m.

Apparently, for the current setup, the different optical flow schemes are closely matched, such that there is no clear winner. Depending on the driven speed it is shown, that the mean velocity errors of all schemes rise with a linear characteristic. Yet, in reference to the total system complexity, that error is relatively small and lies between 6 % and 8 % of the driven ego-velocity. The obtained error curves seem plausible and match our expectations.

Altogether, the good performance of the investigated techniques is reasoned in the fact that the considered scenario is relatively simple. Thus, by changing to

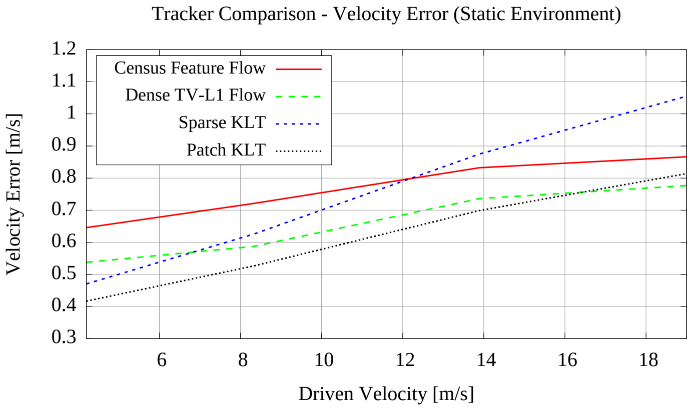
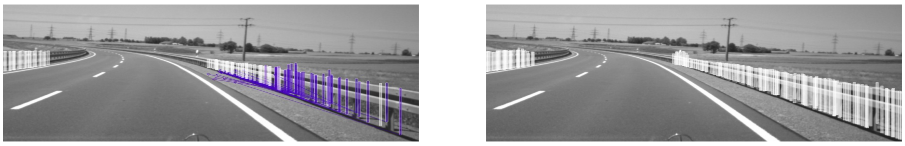


Fig. 11: Direct comparison of the four different tracking schemes. Ideally, the mean absolute velocity should be zero. The quality of the optical flow measurement plays a significant role in this process. Thus, depending on the optical flow scheme, that goal is more or less achieved. For this static urban environment, the differences are rather small.

a highway-like environment, a more challenging scenario is taken into account. It features neither cars nor moving objects but has guard rails on both sides of the road. Naturally, due to their repetitive patterns, guard rails are likely to cause problems for the optical flow computation when driving along in parallel at high speeds. These problems are widely referred to as the *aperture problem* or the *blank wall problem* [7,38]. This is illustrated in Figure 12.



(a) Unreliable optical flow estimates on guard rails lead to wrong Stixel velocity estimates. Additionally, the guard rail is not covered completely.

(b) In contrast, successful optical flow computation allows to obtain correct Stixel velocity estimates. The guard rail is covered much better.

Fig. 12: A precise optical flow estimate is essential for estimating the Stixel motion state reliably. Especially for structures that suffer from aperture problems at high ego velocities, this is a very challenging task. That matter is exemplified with a guard rail scenario.

To increase the degree of difficulty, the ego-velocity is gradually increased to speeds of 8, 14, 20, 28 and 36 m/s. When looking at Figure 12a another important aspect becomes obvious. Problems within the optical flow estimation lead to holes within the line of Stixels covering the guard rail. Typically, this effect is caused by missing or erroneous optical flow measurements. Therefore, in order to draw a more practical conclusion, the performed tests included the completeness measure for the guard rail. This ratio is computed by using ground truth geometry. The corresponding evaluation results are shown in Figure 13.

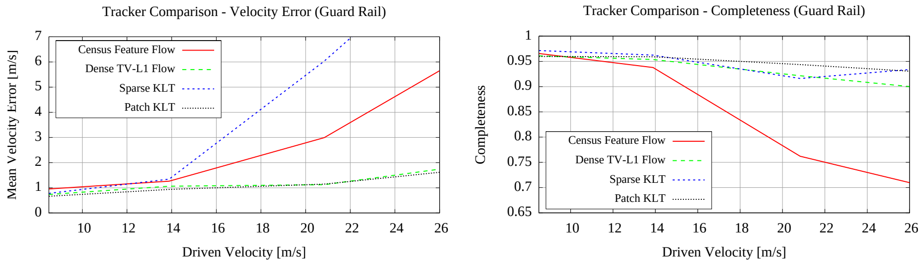


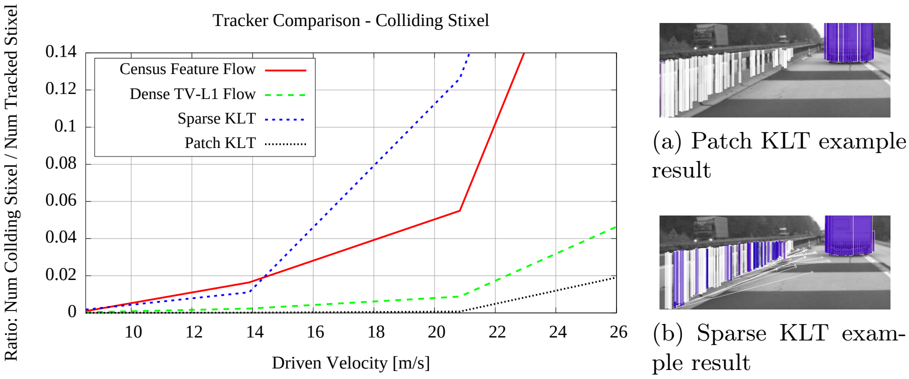
Fig. 13: This figure shows the results of the performance evaluation for the different tracking strategies using the guard rail scenarios. The left figure denotes the remaining mean absolute velocity for the different driven vehicle speeds (8, 14, 20, 28 and 36 m/s). Correspondingly, the right side shows the achieved completeness measure of Stixels covering the guard rail.

Contrary to the previous more static test, the highway environment reveals severe differences between the tracking techniques. Depending on the particular tracking procedure, the velocity estimates as well as the detection rates vary noticeably. The best trade-off with respect to a low velocity error and a satisfying completeness measure is achieved by using the proposed *Patch KLT* procedure or dense *TV-L1* optical flow. Altogether, those two schemes are closely matched. In contrast, the *point feature based KLT* method and the *Census-based* optical flow tracking scheme have serious difficulties estimating the velocity correctly. The *sparse KLT* method yields a high completeness, but its mean absolute estimated velocity is unacceptably high when driving faster than 14 m/s. Even though the *Census-based* feature flow performs slightly better, the achieved velocity estimate is still not good enough to be used in terms of our objectives. Also, that flow scheme has severe problems regarding the detection rate. Thus, when going 14 m/s or faster, that ratio rapidly drops below 75 % completeness.

The good performance of the *TV-L1*-based optical flow is reasoned by the fact, that for every image the assumption of the world to remain static is used as a weak but apparently effective regularization prior for the optical flow estimation. Additionally, the globally optimizing property of TV-L1 supports a solution that is smooth and thus supports our world model too.

With regard to the *Patch KLT*, things are quite similar. The used tracking scheme makes strong use of the Kalman filter prediction as a feed-forward signal. This clearly helps to resolve textural ambiguities of the tracked structures. This way, even though the *sparse feature based KLT* technique allows for the same procedure, things behave somewhat differently. For our understanding, the weakness of the *sparse KLT* method performance results from not considering the change of scale for the feature patch.

The proposed evaluation scheme is practicable as long as there are no moving objects within the scene. With respect to a robustness evaluation on larger datasets it is required to apply other metrics. Therefore, we use the number of colliding Stixels as indicator for tracking errors. Figure 14 demonstrates that the percentage of colliding Stixels correlates perfectly with the mean velocity error presented in the previous section.



(a) Patch KLT example result

(b) Sparse KLT example result

Fig. 14: The diagram shows a comparison of the four tracking approaches (Sparse-KLT, Patch KLT, Census Feature Flow and Dense TV-L1 Flow) in terms of colliding Stixels. The Patch KLT exhibits the fewest tracking errors.

Finally, this allows us to evaluate the robustness of different tracking schemes under various weather conditions.

6.4 Stixel Accuracy

We use the robotic vehicle scenes to assess the Stixel velocity accuracy. The different flow algorithms performed similarly, hence we use the real-time Sparse KLT in the following scenarios. The accuracy of the Stixel measurements was evaluated on 30 robotic scenarios. Therefore, the defined metrics were analyzed within the scope of three different scenario types: Oncoming vehicles, turning maneuver and vehicles passing by.

Velocity Error Robotic vehicles are used to obtain precise ground truth motion data. That data is used for testing the Stixel Kalman filter systems. For this evaluation all dynamic Stixels in the robotic vehicle ROI with an age greater than three frames were used. The resulting weighted mean velocity was compared to the robotic ground truth velocity. Hereby, the goal was to minimize the velocity error for the robotic sequences as well as for the static scenarios. Therefore, more than 20 different filter configurations have been tested.

Figure 15 shows the resulting velocity estimation of an approaching vehicle before and after the optimization process. Both filter configurations perform similarly on the static scenes described in the previous section. The curves illustrate, that in contrast with the optimized filter configuration, the default filter configuration reaches the final velocity approximately 20 m later while exhibiting the same noise level on static scenes. The corresponding qualitative test results are shown in Figure 16.

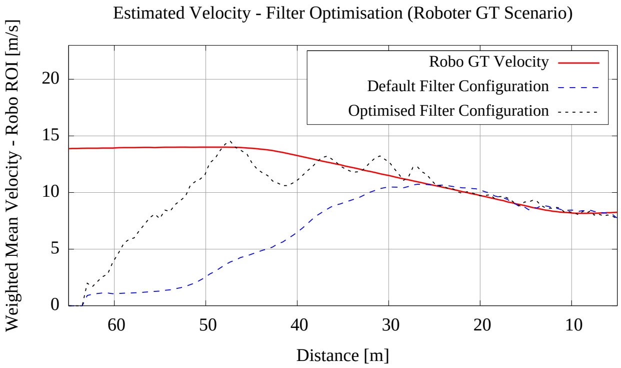


Fig. 15: This figure illustrates the velocity error for an oncoming robotic vehicle (c.f. Figure 16). The diagram shows the Kalman filtered velocity component of two different filter configurations. The ground truth velocity of the robotic vehicle is visualized in red. In contrast to the optimized filter configuration, the default filter configuration reaches the final velocity approximately 20 m later.

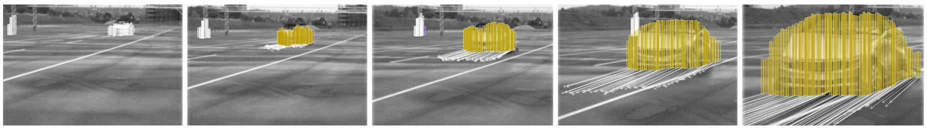


Fig. 16: Sequence of an approaching robotic vehicle. That vehicle starts at a distance of approximately 70 m and closely passes our vehicle to the left.

Distance Error The distance error was evaluated for static and dynamic Stixel measurements on a variety of sequences with approaching robotic vehicles (see Figure 17). With the optimized filter configuration, both the measured and filtered Stixel's distance information averaged over the vehicle front yield congruent output.

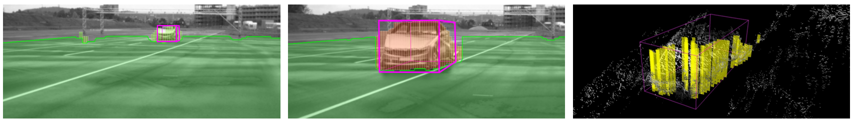


Fig. 17: Example images for the distance error evaluation. The calculated robotic vehicle position (marked in magenta) is projected into the image plane and used for collecting all the Stixel measurements representing the vehicle's front. On the right a 3D representation of the scene is visualized.

Figure 18 depicts the mean distance error to our robotic ground-truth distance of all static Stixels representing the front of the robotic vehicle. The second axis shows the calculated standard deviation for these Stixel measurements added on the mean error.

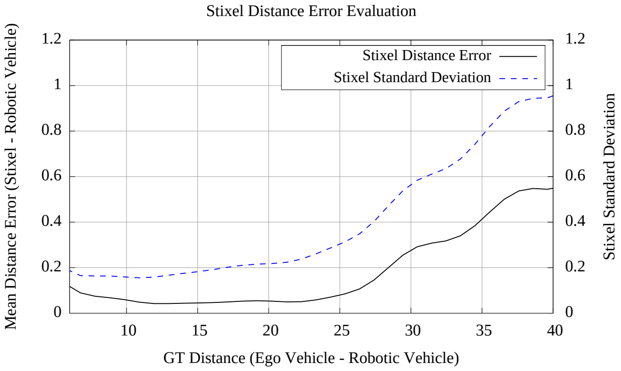


Fig. 18: The diagram shows the mean distance error of all Stixel measurements that are located at the front of the vehicle. The error increases in greater distances due to measurement noise as well as in the close-up range. The latter is explained by the violated vertical pose constraint on the engine hood. The 3D representation in Figure 17 visualizes the displaced Stixel position.

The curves show a noisy depth estimation at larger distances as well as an increasing distance error in the close-up range ($< 15\,m$). The measurements near the ego vehicle address the violated vertical pose constraint described in

Section 4. As a consequence, if the vehicles front with its engine hood is modeled end-to-end by one Stixel, its position in z direction will be displaced towards the windshield (see Figure 17). That means, the Stixel measurements are seen as too far away. More details on the static Stixel position accuracy can be found in [31].

7 Conclusion

In this research we presented an evaluation framework for stereo-based driver assistance that operates on large image data bases and demands very little groundtruthing effort. To show the power of the evaluation framework, we performed evaluations on several stereo algorithms where we found the Semi-Global Matching (SGM) to be the best performing stereo algorithm on pixel-level, on freespace level and on object level. For the intermediate representation, the Stixel World, we detected Stixel phantoms only for challenging weather scenarios. By using the evaluation framework, the phantom rate could be further reduced by a factor of three while maintaining the detection rate of the Stixel representation. Comparing four optical flow algorithms used to generate dynamic Stixels we found the Patch KLT to be the best performing algorithm under the aspects of accuracy and robustness. For the absolute Stixel accuracy we determined a $0.5\,m$ position error at $40\,m$ distance using data from robotic vehicles as reference.

For future work we will extend this analysis framework to all vision-based driver assistance algorithms currently under development, to obtain meaningful performance figures. In addition, we consider making parts of the used data publicly available as part of a challenge that specifically addresses 3D outdoor scene analysis under all weather conditions.

References

1. Hernán Badino. A robust approach for ego-motion estimation using a mobile stereo platform. In 1^{st} *International Workshop on Complex Motion, IWCM*, Günzburg, Germany, October 2004. Springer.
2. Hernán Badino, Uwe Franke, and Rolf Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Workshop on Dynamical Vision, ICCV*, Rio de Janeiro, Brazil, October 2007.
3. Hernán Badino, Uwe Franke, and David Pfeiffer. The Stixel World - A compact medium level representation of the 3D-world. In *German Association for Pattern Recognition (DAGM)*, pages 51–60, Jena, Germany, September 2009.
4. Alexander Barth. *Vehicle Tracking and Motion Estimation Based on Stereo Vision Sequences.* PhD thesis, Friedrich-Wilhelms-Universitaet zu Bonn, September 2010.
5. Alexander Barth and Uwe Franke. Where will the oncoming vehicle be the next second? In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1068–1073, Eindhoven, Netherlands, April 2008.
6. Alexander Barth, Jan Siegemund, Uwe Franke, and Wolfgang Förstner. Simultaneous estimation of pose and motion at highly dynamic turn maneuvers. In *German Association for Pattern Recognition (DAGM)*, pages 262–271, Jena, Germany, September 2009. Springer.

7. Thomas Brox and Joachim Weickert. Nonlinear matrix diffusion for optic flow estimation. In *German Association for Pattern Recognition (DAGM)*, pages 446–453, Zürich, Switzerland, September 2002.
8. R. Collins, Y. Tsin, J.R. Miller, and A. Lipton. Using a dem to determine geospatial object trajectories. In *Proceedings of the 1998 DARPA Image Understanding Workshop*, pages 115–122, 1998.
9. Patrick Courtney, Neil Thacker, and Adrian Clark. Algorithmic modeling for performance evaluation. In *IAPR Conference on Machine Vision Applications (MVA)*, pages 219–228, 1997.
10. P. Dreuw, P. Steingrube, T. Deselaers, and H. Ney. Smoothed disparity maps for continuous american sign language recognition. In *Iberian Conference on Pattern Recognition and Image Analysis, Povoa de Varzim, Portugal*, 2009.
11. B. R. Duffy, C. Garcia, C. F. B. Rooney, G.M.P. O Hare, and G. M. P. O Hare. Sensor fusion for social robotics. In *31st International Symposium on Robotics*, pages 155–170, 2000.
12. M. Everingham, A. Zisserman, C.K.I. Williams, , and L. Van Gool. The 2005 pascal visual object classes challenge. In *Selected Proceedings of the 1st PASCAL Challenges Workshop*. Springer, 2006.
13. Uwe Franke. Real-time stereo vision for urban traffic scene understanding. In *IEEE Intelligent Vehicles Symposium (IV)*, 2000.
14. Uwe Franke, Clemens Rabe, Hernán Badino, and Stefan Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In *German Association for Pattern Recognition (DAGM)*, Vienna, Austria, September 2005.
15. Friedrich Fraundorfer, Davide Scaramuzza, and Marc Pollefeys. A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1899–1904, Anchorage, Alaska, USA, May 2010.
16. Stefan Gehrig, Felix Eberli, and Thomas Meyer. A real-time low-power stereo vision engine using semi-global matching. In *International Conference on Computer Vision Systems (ICVS)*, Liège, Belgium, October 2009. Springer-Verlag.
17. Andree Hohm, Christian Wojek, Schiele Bernt, and Hermann Winner. Multi level sensorfusion and computer-vision algorithms within a driver assistance system for avoiding overtaking accidents. In *FISITA World Automotive Congress*, pages 1–14, 2008.
18. Tsai Hong, Tommy Chang, Ayako Takeuchi, Gerry Cheok, Harry Scott, and Michael Shneier. Performance evaluation of sensors on mobile vehicles using a large data repository and ground truth. In *Proceedings PerMIS*, 2003.
19. Weihua Huang, Chew Lim Tan, and Jiuzhou Zhao. Generating ground truthed dataset of chart images: Automatic or semi-automatic? In *GREC 07*, pages 266–277, 2007.
20. iMAR Navigation. iTraceRT-F200. `http://www.imar-navigation.de/`, August 2011.
21. Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 486–492, San Diego, CA, USA, June 2010.
22. Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision (IJCV)*, 74(3):343–364, 2007.
23. Anat Levin and Richard Szeliski. Visual odometry and map correlation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 611–618, Washington, DC, USA, June 2004.

24. Zhifeng Liu and Reinhard Klette. Approximated ground truth for stereo and motion analysis on real-world sequences. In *Proceedings "PSIVT 2009", LNCS 5414.*, 2009.
25. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Seventh International Joint Conference on Artifical Intelligence, IJCAI 1981*, pages 674–679, Vancouver, Canada, 1981.
26. Vasant Manohar, Padmanabhan Soundararajan, Harish Raju, Dmitry Goldgof, Rangachar Kasturi, and John Garofolo. Performance evaluation of object detection and tracking in video. In *ACCV*, volume 3852 of *Lecture Notes in Computer Science*, pages 151–161. Springer, 2006.
27. Vladimir Y. Mariano, Junghye Min, Jun Hyeong Park, Rangachar Kasturi, David Mihalcik, Huiping Li, D. S. Doermann, and T. Drayer. Performance evaluation of object detection algorithms. In *International Conference on Pattern Recognition (ICPR)*, 2002.
28. Sandino Morales, Tobi Vaudrey, and Reinhard Klette. Robustness evaluation of stereo algorithms on long stereo sequences. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 347 – 352, 2009.
29. Thomas Müller, Jens Rannacher, Clemens Rabe, and Uwe Franke. Feature and depth-supported modified total variation optical flow for 3d motion field estimation in real scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1193–1200, Colorado Springs, CO, USA, June 2011.
30. David Pfeiffer and Uwe Franke. Efficient representation of traffic scenes by means of dynamic Stixels. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 217–224, San Diego, CA, USA, June 2010.
31. David Pfeiffer, Sandino Morales, Alexander Barth, and Uwe Franke. Ground truth evaluation of the Stixel representation using laser scanners. In *IEEE Conference on Intelligent Transportation Systems (ITSC)*, Maideira Island, Portugal, September 2010.
32. Daniel Scharstein and Richard Szeliski. Middlebury online stereo evaluation, 2002. http://vision.middlebury.edu/stereo.
33. Nicolai Schneider. Evaluation of stereo-based scene analysis under real-world conditions. Master's thesis, Brunel University, July 2011.
34. Continental Automotive Industrial Sensors. ARS 300 Long Range Radar Sensor 77 GHz. `http://www.conti-online.com/generator/www/de/en/continental/industrial_sensors/themes/ars_300/ars_300_en.html`, July 2011.
35. Fridtjof Stein. Efficient computation of optical flow using the census transform. In *German Association for Pattern Recognition (DAGM)*, pages 79–86, Tübingen, Germany, August 2004.
36. Pascal Steingrube, Stefan Gehrig, and Uwe Franke. Performance evaluation of stereo algorithms for automotive applications. In *International Conference on Computer Vision Systems (ICVS)*, pages 285–294, Liège, Belgium, October 2009. Springer-Verlag.
37. Tech-News. Toyota' lexus ls 460 employs stereo camera, viewed 2009/04/15. http://techon.nikkeibp.co.jp/english/NEWS_EN/20060301/113832/.
38. Massimo Tistarelli. Multiple constraints for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 61–70, Stockholm, Sweden, May 1994.
39. Carlo Tomasi and Jianbo Shi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Los Alamitos, CA, USA, June 1994.

40. Toby Vaudrey, Clemens Rabe, Reinhard Klette, and James Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *Proceedings PSIVT 2009, LNCS 5414.*, 2009.
41. Kunio Yamada, Kenji Mochizuki, Kiyoharu Aizawa, and Takahiro Saito. Motion segmentation with census transform. In *Advances in Multimedia Information Processing, Second IEEE Pacific Rim Conference on Multimedia*, volume 2195, pages 903–908, Bejing, China, October 2001. Springer.
42. A. Yilmaz. Sensor fusion in computer vision. In *EEE GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, 2007.
43. Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l1 optical flow. In *German Association for Pattern Recognition (DAGM)*, pages 214–223, Heidelberg, Germany, September 2007.
44. Richard Zanibbi, Dorothea Blostein, and James R. Cordy. White-box evaluation of computer vision algorithms through explicit decision-making, 2009.